Metric Data: Clustering

Sudipto Guha
UPENN

(Semi) Metric spaces

- A set of points X
  - $D(x_i,x_j) \geq 0$
  - $D(x_i,x_i)=0$
  - $D(x_i,x_j)=D(x_j,x_i)$
  - $D(x_i,x_j)+D(x_j,x_k) \geq D(x_i,x_k)$
- How do we store n^2 numbers?

Oracle Distance Functions

- You do not.
- Given x_i,x_j compute $d(x_i,x_j)$
- Or better: Is $D(x_i,x_j) \leq r$
- Examples?
- How do you verify that the oracle is giving you a metric?

The Assumptions

- You can not.
- Many algorithms rely on the assumption for running time …
- Unintended Consequences: The only points your algorithm uses must belong to the input.

The Plan

- Clustering …
- But
  - What is clustering ?

You tell me …

- An endless list
  - Are you clusters fat?
  - Do they drink coffee?
  - Do they have central points ? Metric) Sphere
  - Does a point belong to one cluster only?
  - Do we cluster all the points all the time?
- This talk: yes, yes & yes.
- But these are by no means the only choices.
- Then again what CAN you do in a streaming setting?

Clustering in a streaming model

- We can only store small amount of information) Central points
  - Each point assigned to its nearest point
- Sliding windows, clustering 1-ε fraction …
- What is the measure of clustering?

Two Problems

- Cluster centers: $S=\{s_1,s_2,\ldots s_k\}$
- Covering by discs:
  - K center: $\text{Min}_S \max_i \min_j D(x_i,S_j)$
- More robust measure
  - K median: $\text{Min}_S \sum_i \min_j D(x_i,S_j)$

K Center

- Charikar, Chekuri, Feder, & Motwani `97
- O(k) space, factor 8 approximation.
  - (More later)
- Statutory warning: NP hard to approximate better than 2
  - Points to ponder: Does that imply anything for the streaming model? What is k?
  - Any algorithm achieving a factor better than 2 must store all the points.

An offline algorithm

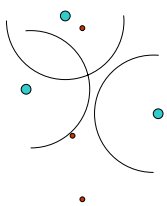
- Hochbaum Shmoys `85
- Gonzales `85
  - Pick the first point as s_1, $S=\{s_1\}$
  - Pick the point farthest from S
  - Repeat the above k-1 times.
- Why does this work?

Approximation Algorithms: Quest for Lower Bounds

- Run the algorithm for one more step and get k+1 points, S'=S [{x}
- Let r=smallest pairwise distance in S'
- Claim OPT ¸ r/2
  - If not then OPT must have k+1 centers…
- Observe: all points are distance r from S
- **Witness of a lower bound of r**

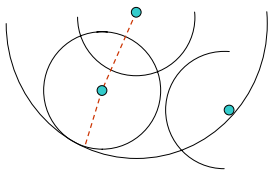
A First Cut

- Maintain k points at distance r from each other
- New point arrives
  - Close: Ignore
  - Far: Merge closest pair
- Does not work
- Error builds up.



The ultimate truth: Doubling?

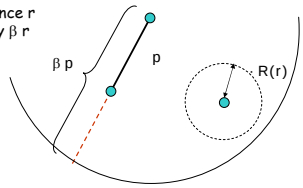
- Grow more …



- In particular grow by β factor

Proof

- Invariant
  - k+1 points at distance r
  - Keep points sep. by β r
  - R(r) radius
- New bound: p
- p ¸ β r
- R(p)=β p + R(r)
  =β p + R(t/β)
  =β^2p/(β-1)
- β=2; R(p)=4p
- OPT ¸ p/2



Other properties

- Clusters, once merged stay together.
- An incremental/online model
  - A small space online algorithm is a streaming algorithm.
- Randomization) 5.4 (choose β at random)
- Diameter measure, clique partitions…

Epilogue: K center

- Note
  - 2+ε using $O(k\varepsilon^{-1} \log \Delta)$ space is trivial.
  - 2+ε using $O(k\varepsilon^{-1})$ space and 2 passes is also trivial.
- Δ is the ratio of min to max distance) Precision parameter
- Dependence on precision is highly avoidable (whenever possible) in geometric problems.
- Guha, Khuller, McGregor `xx
  - 2+ε approximation using $O_\varepsilon(k)$ space, single pass

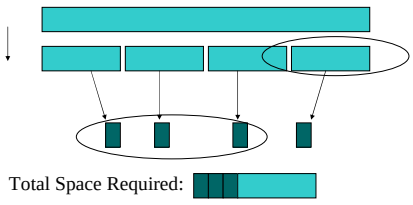
K median

- Guha, Mishra, Motwani, & O'callaghan '00
- Meyerson '01
- Charikar, Panigrahy, & O'callaghan '03

1. $O(n^\varepsilon k)$ space $O(2^{1/\varepsilon})$ approx
2. $O(k \log^2 n)$ space $O(1)$ approx

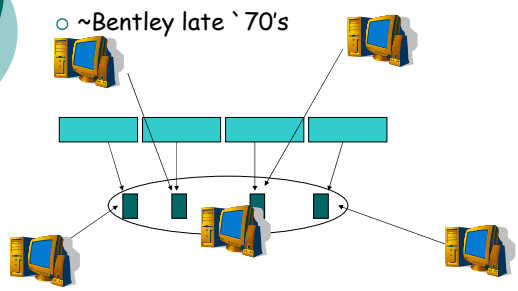
Divide & Conquer/Merge-Reduce

- ~Bentley late `70's



Divide & Conquer/Merge-Reduce

- ~Bentley late `70's

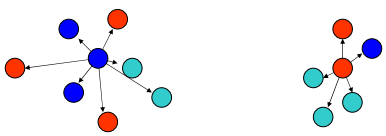


Summarize a pointset for clustering?

- Cluster, of course ☺
- $(n/k)^{1/2}$ pieces, size $(nk)^{1/2}$ each.
- Each piece sends the k medians
  - (more on this)
- Cluster the $(nk)^{1/2}$ medians

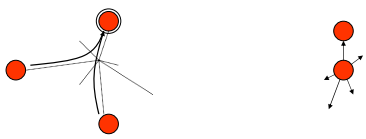
Consider 3 partitions

- Arbitrary, represented by the colors



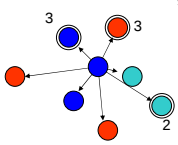
Consider 3 partitions

- Consider just the reds
- 9 a clustering of twice its share



Consider 3 partitions

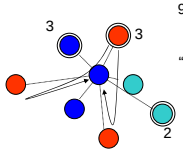
- We should count importance …



9 clustering 2 OPT
) 2α OPT soln

Consider 3 partitions

- We should count importance …

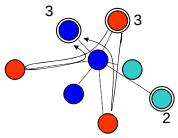


9 clustering 2 OPT
) 2α OPT soln

"Send Back" & use fictitious center
) 9 (1+2α) OPT

Consider 3 partitions

- We should count importance …



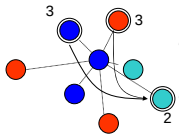
9 clustering 2 OPT
) 2α OPT soln

"Send Back" & use fictitious center
) 9 (1+2α) OPT

"Send Back" & use a point we have
) 9 2(1+2α) OPT

Consider 3 partitions

- We should count importance …



9 clustering 2 OPT
) 2α OPT soln

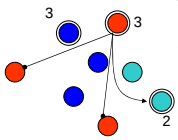
"Send Back" & use fictitious center
) 9 (1+2α) OPT

"Send Back" & use a point we have
) 9 2(1+2α) OPT

Cluster of clusters
) 2α(1+2α) OPT

Consider 3 partitions

- We should count importance …



9 clustering 2 OPT
) 2α OPT soln

"Send Back" & use fictitious center
) 9 (1+2α) OPT

"Send Back" & use a point we have
) 9 2(1+2α) OPT

Cluster of clusters
) 2α(1+2α) OPT

Total) (2α + 2α(1+2α)) OPT

That's it really!

- Did not need exactly k centers in intermediate step

9 clustering 2 OPT
) 2β OPT soln

"Send Back" & use fictitious center
) 9 (1+2β) OPT

"Send Back" & use a point we have
) 9 2(1+2β) OPT

Cluster of clusters
) 2α(1+2β) OPT

Total) (2β+ 2α(1+2β)) OPT

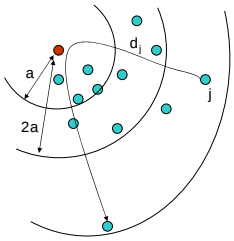
A dash of facilities

- Facility location Problem
  - There is a cost f_j of building a "center" at node j.
- Another day, another adventure…
- This talk: Imagine that the number of clusters are not fixed
  - Cost of each cluster is f
  - Minimize sum of distances plus the cost of clusters

A simple algorithm

- Meyerson '01
- Maintain a set of centers S
  - New point x_j
  - $D(x_j,S) = \delta_j$
  - Declare x_j as a center with prob. δ_j/f
  - OW x_j assigned to nearest in S

Analysis



a=Average radius…

Rings in powers of 2

Focus on a specific ring

Expected assignment cost *before 1st center is opened:* f (why?)

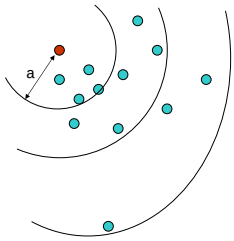
Any node pays at most thrice its contribution to OPT

Cost=2f (log n)

$+ \sum_j [\ 3d_j + f(3d_j)/f]$

Something's Missing!

Analysis



Node pays $a+d_j+f(a+d_j)/f$

Summed over:

2f + 2OPT + 2Self-contib to OPT

2f (log n) + 6(Self-contrib OPT)

Per cluster …

Net: 2f(1+log n) (k)+ 8OPT

Enter the Magic Birdie

- Tells you OPT
- You set
  - f=OPT/(k log n)

- (Expected)
  - Net cost = 10 OPT
  - Number of Medians = 10k log n
- Run O(log n) times in parallel

If you hate birdies…

- Easy solution: Guess OPT
  - log Δ guesses
  - Not very desirable
- Use previous k-center approach of
  - keeping a bound
  - increasing by β
  - Showing contributions do not add up
  - Desirable (the contrib of CPO in this context)
  - Lots of equations ☹

Recap

- A Metric Space assumption provides a surprising amount of information.
- We can cluster in streaming model
  - Geometric Phases
  - Need
    - Small Witnesses
    - Decomposition Theorems
  - Open Q. Is the randomization necessary?

Time (not) for …